

## **Comp3330/6380 Machine Intelligence, Semester 1, 2013**

### **Homework Assignment 2 (HWA 2): Intelligent System Design**

Deadline: Week 11 (26.5.2013, midnight)

Maximum possible marks: 10

#### **Description**

This assignment is about intelligent system design for the popular web game 2048.

Teams of (1-4) students will develop players for the game 2048.

The mark for the assignment (maximally 10 marks) will be determined from two components:

1. A technical report that describes how your system works. The recommended length of the report is about 4-10 pages. Aim at describing and discussing your system clearly and concisely following instruction of the individual questions below.
2. Performance evaluation of your player. Include all files in your submission that are required for running your system following instruction of the individual questions below.

This assignment can be done in teamwork with other students from this class (1-4 people per team) and we encourage you to do this. A completed electronic ACS must accompany all assessment submissions. Any additional help that you use has to be explicitly acknowledged in your submission.

Please submit your assignment electronically via the assignment section in blackboard. Include your report and all relevant software. Please let us know any questions or if anything requires further clarification.

#### **Useful Background Literature**

A previous AI written for the game in javascript is available here: <https://github.com/ov3y/2048-AI>

It uses alpha-beta pruning to build a possible game tree (this is similar to techniques used for AI based chess and Go).

The game Tic-Tac-Toe is mechanically solvable using matchboxes and beans to indicate learning statistics. This is achieved by reducing the number of possible play positions using reflections and rotations. The result is similar to TD-learning. An article about the machine is available here: <http://makezine.com/2009/11/02/mechanical-tic-tac-toe-computer/>

## Q1 Intelligent Agents Literature Review and Discussion (3 marks)

Students are required to review previous AI solutions and literature for the game 2048 and compare to AI solutions for other similar games. The report should include a discussion of the development of game-based AI. Students should also investigate current challenges in the field such as developing an AI for Go, and discuss why these types of challenges are important for developing future thinking systems.

## Q2 2048 AI (7 marks)

Students will develop and document a successful algorithm to solve the game 2048. This will be based on one or more of the techniques covered in lectures, extended to fit the problem. Marks will be awarded for:

- Clear description of the learning or search algorithm(s) used in your approach (1 mark).
- Discussion of the process for tuning these particular processes and developing appropriate representations of the environment (2 marks).
- Explanation of the features of the problem which make the solution appropriate (2 marks)

Please include high level pseudocode for the algorithm(s) used, including any changes from standard algorithms that you have made.

- Students will also submit a working agent (with execution instructions) to demonstrate the outcome of their AI. This will be marked on performance in several games with undisclosed random seeds. Marks will be awarded based on the overall score of the agent, the ability to achieve the 2048 tile, and the overall quality of application of the chosen learning algorithm to the game. (2 marks)

## Interfacing to 2048

The 2048 game was ported to python in order to work with Pybrain. We have added some features which make it easier to use with pybrain and other learning environments. We provide the game code as ported, modified slightly to give more information back, and a pybrain task environment which can be used in training experiments. Example code is provided to train a basic agent, but it will have a low success rate as many parts of the implementation can be improved.

Students may modify any of the files included in the homework release except "game.py". "PybrainEnvironment.py" handles translation to the pybrain library, and can be used to enhance the representation of the game state and similar changes to improve the agents' perception of the problem. Alternatively, students may use another learning or search strategy altogether (however we can't guarantee code/debugging support for other environments). Students' marks will be based on the ability to apply concepts covered by the Comp3330 lectures to this challenge in a way which shows understanding of the problem and the material covered in the lectures. If students have difficulty installing or running certain parts of the software, please ask questions either via email or in the labs.